



Calhoun: The NPS Institutional Archive

Faculty and Researcher Publications

Faculty and Researcher Publications

2000

Current Developments in Underwater Vehicle Control and Navigation: The NPS ARIES AUV

Healey, A. J.



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

Current Developments in Underwater Vehicle Control and Navigation: The NPS ARIES AUV

David B. Marco
Dept. of Mechanical Engineering
Naval Postgraduate School
Monterey, CA

Anthony J. Healey
Dept. of Mechanical Engineering
Naval Postgraduate School
Monterey, CA

Abstract-This paper provides an overview of the Naval Postgraduate School ARIES autonomous underwater vehicle. An attempt is made to highlight its' current operational capabilities and provide a description of future enhancements for greater mission utility and flexibility. An overview of the vehicle design along with descriptions of all major hardware components and sensors is given. A major discussion of the implementation of a modular, multi-rate, multi-process software architecture for the ARIES is provided. The architecture is designed to operate using a single computer processor or two independent, cooperating processors linked through a network interface for improved load balancing. A dual computer implementation is presented here since each processor assumes different tasks for mission operation. Also included is a section on the underwater navigation method used. It involves the use of a real-time extended Kalman filter that fuses all sensor data and computes the real time position, orientation, velocity, etc., of the vehicle. Issues of navigational accuracy of the filter are also discussed. The work concludes with a discussion of using the ARIES as a communications server in a multi-vehicle environment. It is proposed to use the ARIES as a mobile communications relay between a command and control station on the surface and multiple vehicles operating below. The advantages of using a mobile relay over fixed buoys are discussed.

I. INTRODUCTION

The Naval Postgraduate School Center for AUV Research has been building, operating, and researching autonomous underwater vehicles (AUVs) since 1987. Each new generation of vehicles have substantially increased operational capabilities and are much more robust and sophisticated in terms of hardware and computer software. These vehicles have also moved from operating in swimming pool environments to the open ocean. This paper will present a description of the latest generation of underwater vehicle named the NPS ARIES AUV. A photograph of the vehicle is shown in Figure 1 and Figure 2 shows the component layout of the vehicle. The hull was

outfitted in the fall of 1999 and has recently become fully operational (Spring 2000), and at the present time, only software enhancements are required. The vehicle has been designed as a network server platform/target reacquisition vehicle, and has been operated during AUVfest '99 in Gulfport, MS (AUVFest '99). Currently, the vehicle operates regularly in Monterey.

II. VEHICLE DESCRIPTION

Descriptions of the major hardware components of the ARIES are given below.

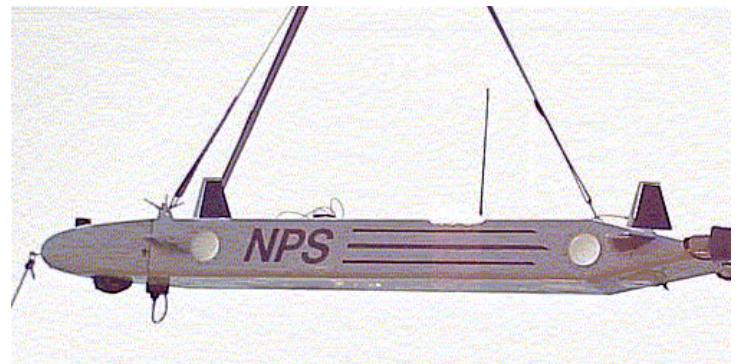


Figure 1. The NPS ARIES AUV on the hook at AUVFest '99

Dimensions and Endurance: The vehicle weighs 225 Kg and measures approximately 3 m long, 0.4 m wide and 0.25 m high. The hull is constructed of 1/4" thick 6061 aluminum and forms the main pressure vessel that houses all electronics, computers, and batteries. A flooded fiberglass nose is used to house the external sensors and power on/off switches and status indicators. It is capable of a top speed of 3.5 knots and is powered by six 12 volt rechargeable lead acid batteries. The endurance is

approximately 4 hours at top speed, 20 hours hotel load only. The ARIES was primarily designed for shallow water operations and can operate safely down to 30 meters. However, with hull strengthening in certain areas, a depth of 100 meters may be attained.

Propulsion and Motion Control Systems: Main propulsion is achieved using twin ½ Hp electric drive thrusters located at the stern. During normal flight, heading and depth is controlled using upper bow and stern rudders and a set of bow planes and stern planes. Since the control fins are ineffective during very slow or zero forward speed maneuvers, vertical and lateral cross-body thrusters are used to control surge, sway, heave, pitch, and yaw, motions.

Navigation Sensors: The sensor suite used for navigation includes a 1200 kHz RD Instruments Navigator DVL that also contains a TCM2 magnetic compass. This instrument measures the vehicle ground speed, altitude, and magnetic heading. Angular rates and accelerations are measured using a Systron Donner 3-axis Motion Pak IMU. While surfaced, carrier phase differential GPS (DGPS CP) is available to correct any navigational errors accumulated during the submerged phases of a mission.

Sonar and Video Sensors: A Tritech ST725 scanning sonar or an ST1000 profiling sonar is used for obstacle avoidance and target acquisition/reacquisition. The sonar heads can scan continuously through 360° of rotation or swept through a defined angular sector. A fixed focus wide-angle video camera is located in the nose and connected to a DVC recorder. The computer is interfaced to the recorder and controls on/off and start/stop record functions. While recording, the date, time, vehicle position, depth and altitude is superimposed on the video image.

Vehicle/Operator Communications: Radio Modems are used for high bandwidth command, control, and system monitoring while the vehicle is deployed and surfaced. While submerged, an acoustic modem is used for low bandwidth communications. In the laboratory environment, a high-speed thin-wire ethernet connection is used for software development and mission data upload/download.

III. COMPUTER HARDWARE ARCHITECTURE

A photograph of the dual computer system unit is shown in Figure 3 and measures approximately 28 x 20 x 20 cm. It consists of two Ampro Little Board 166 MHz Pentium computers with 64 MB RAM, four serial ports, a network interface, and a 2.5 GB hard drive each. Two DC/DC voltage converters for powering both computer systems and peripherals are integrated into the computer package. The entire computer system draws a nominal 48 Watts.

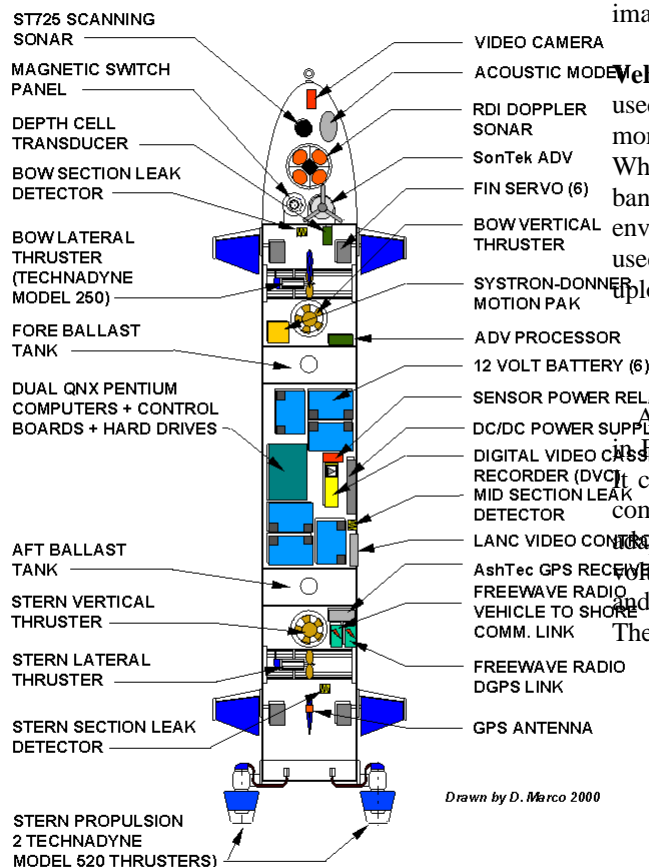


Figure 2. Hardware Components of the NPS ARIES

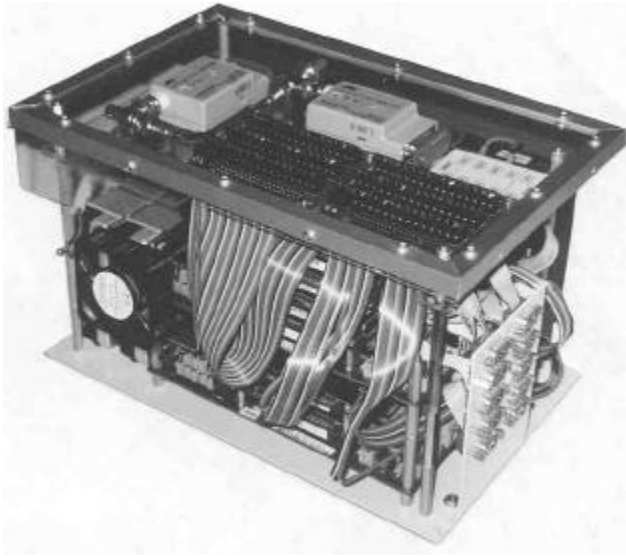


Figure 3. Dual Computer System Unit

Both systems use TCP/IP sockets over thinwire ethernet for inter processor communications and connections to an external LAN. The sensor data gathering computer is designated QNXT, while the second is named QNXE and executes the various auto-pilots for servo level control. Both computers are used as the baseboard for a stack of Diamond Systems PC-104 data acquisition boards. Four boards are associated with system QNXT and are the Diamond-MM 12 bit A/D converter, Diamond-MM-16 16 bit A/D converter, Topaz-MM high current TTL card, and the Emerald-MM 4 port RS-232 serial card. These boards are used for sensor control and data gathering. Two boards are used with the system QNXE and are the Ruby-MM 12 bit D/A converter and the Quartz-MM 10 Channel timer card. And are used for servo control of the fins and thrusters.

IV. COMPUTER SOFTWARE ARCHITECTURE

A. The Architecture

A diagram outlining the modular, multi-rate, multi-process software architecture is shown in Figure 4. The architecture is designed to operate using a single computer processor or two independent, cooperating processors linked through a network interface. Splitting the processing between two computers can significantly improve computational load balancing and software segregation. A dual computer implementation will be presented here, since in the ARIES, each processor assumes different tasks for mission operation.

Both computers run the QNX real time operating system using synchronous socket sender and receiver network

processes for data sharing between the two. Inter-process communication is achieved using semaphore controlled shared memory structures. At boot time, the network processes are started automatically and all shared memory segments are created in order to minimize the amount of manual setup performed by the user.

All vehicle sensors are interrogated by separate, independently controlled, concurrent processes, and there is no restriction on whether the processes operate synchronously or asynchronously. Since various sensors gather data at different rates, each process may be tailored to operate at the acquisition speed of the respective sensor. Each process may be started, stopped, or reset independently allowing easy reconfiguration of the sensor suite needed for a given mission. All processes are written in the C programming language.

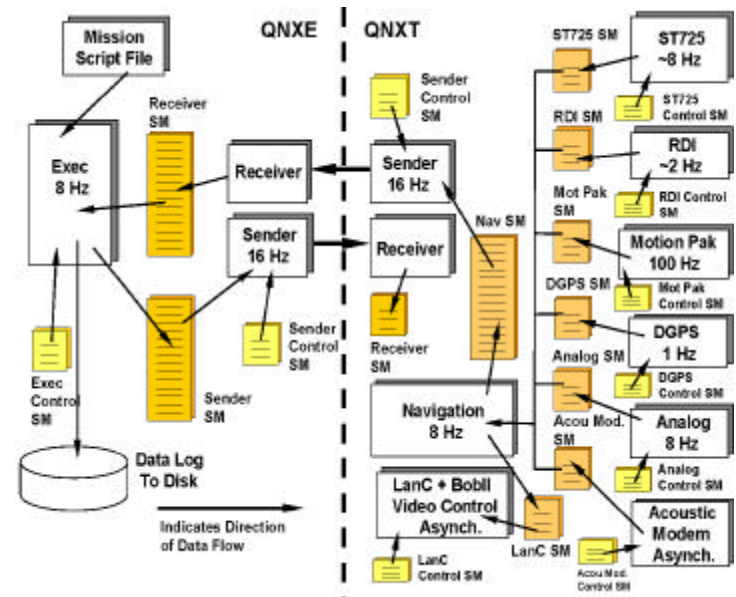


Figure 4. Dual Computer Software Architecture

To allow synchronous sensor fusion, each process contains a unique shared memory data structure that is updated at the specific rate of each sensor. All sensor data are accessible to a synchronous navigation process through shared memory and is a main feature of the software architecture proposed. An example of a typical sensor shared memory structure is shown below.

```
typedef struct {
    int Proc_Counter;
    ...
    Sensor Data
    ...
    int Proc_Error;
    sem_t Proc_Semaphore;
} data_structure;
```

The process counter, *Proc_Counter* is a value incremented each time the sensor is read. If this value fails to increase through time, the Navigator will detect this and assume the sensor controlling process has died or is “hung up”. If the controlling process detects the sensor is not operating properly, the error flag *Proc_Error* is set to *TRUE* and the navigator will detect this and take the appropriate actions. To prevent data corruption, a semaphore, *Proc_Semaphore* is used and will disallow simultaneous data read/writes from competing processes.

Incorporated into the navigation process is an extended Kalman filter that fuses all sensor data and computes the real time position, orientation, velocity, etc., of the vehicle. The dual computer implementation uses one processor for data gathering and running the navigation filters, while the second uses the output from the filters to operate the various auto-pilots for servo level control. Once the state information is computed, it is transmitted to the second computer over standard TCP/IP sockets.

B. Mission Control Modes

At the present time all vehicle behaviors are determined by a preprogrammed mission script file that is parsed in the QNXE computer by the process *Exec*. The file contains a sequential list of commands that vehicle is to follow during a mission. These commands may be as simple as setting the stern propulsion thruster speeds to more complex maneuvers such as commanding the vehicle to repeatedly fly over a submerged target at a given GPS coordinate using altitude and cross track error control.

Below is an example of a simple mission script file.

```
SET_ALTITUDE                2.0
SET_HEADING                 60.0
SET_STERN_THRUSTER_SPEED    700.0
USE_ALTITUDE_CONTROL
USE_HEADING_CONTROL
USE_STERN_THRUSTER_SPEED_CONTROL
SET_FLIGHT_DURATION         300.0
SHUTDOWN
```

This commands the vehicle to fly above the bottom at an altitude of 2 meters with a heading of 60 degrees, while running the twin stern thrusters at 700 rpm. The run is designed to last for a total of 300 seconds. The above mission could also have used depth instead of altitude control by simply replacing *SET_ALTITUDE* with *SET_DEPTH* and *USE_ALTITUDE_CONTROL* with *USE_DEPTH_CONTROL*.

The above auto-pilots can be useful for certain missions where exact spatial control is not important. However today, many missions require accurate navigation and

control between waypoints for the purposes of target acquisition and reacquisition with data collected using acoustic sensors or video cameras. This may be achieved through the use of a cross-track error controller. It allows the vehicle to track a straight line between waypoints even in the presence of currents.

At this time, cross-track error control is used for three separate modes:

1. Pattern Search
2. Target Flyby
3. Target Hovering

The first mode is Pattern Search and may be commanded in the mission script file using the following:

```
.
.
USE_CTE_CONTROL
USE_PATTERN_SEARCH PatternFile.inp
.
.
```

The file *PatternFile.inp* contains waypoints previously generated and may be standard “lawn mower”, “box”, or any other pattern design the user may wish. Other set points are also included in the pattern file such as commanded depth or altitude above bottom.

Target Flyby control is the second mode and is issued by:

```
.
.
USE_CTE_CONTROL
USE_FLYBY_CONTROL
SET_FLYBY_TARGET 3636.2354 N 12148.78654 W 3 60.0
SET_FLYBY_TARGET 3637.9876 N 12148.78654 W 2 120.0
.
.
```

The above instructs the vehicle to fly over the area located at coordinates 3636.2354 N 12148.78654 W three times at an approach heading of 60 degrees. After this maneuver has completed, the vehicle will fly over the second set of coordinates 2 times at a heading of 120 degrees.

Target Hover control is the third and final mode available. It involves commanding the vehicle to transit to specific coordinates of interest and slow to a stop over the area for a certain amount of time. Dynamic positioning over the target will be achieved using the cross-body thrusters along with the stern thrusters. Below is a section of the script file for this purpose.

```
.
.
USE_CTE_CONTROL
USE_HOVER_CONTROL
SET_HOVER_TARGET 3636.2354 N 12148.78654 W 60.0 40.0
SET_HOVER_TARGET 3637.9876 N 12148.78654 W 100.0 200.0
.
.
```

The above commands the vehicle to transit to coordinate 3636.2354 N 12148.78654 W, hover for 60 seconds at a

heading of 40 degrees. Once this is complete, the vehicle moves to the second set of coordinates and hovers for 100 seconds at a heading angle of 200 degrees. It should be noted that when using the previous two control modes the commanded depth or altitude may be set using either SET_DEPTH or SET_ALTITUDE.

V. NAVIGATION

The ARIES vehicle uses an INS / DOPPLER / DGPS navigational suite and an Extended Kalman Filter (EKF) (Healey, An, Marco, 1998) and may be tuned for optimal performance given a set of data. The main impediments to navigational accuracy are the heading reference and the speed over ground measurement. In this system, the heading reference is derived from both the compass located in the RDI Navigator and the Systron Donner IMU, which provides yaw rate. The fusion of the yaw rate and the compass data leads to an identification of the yaw rate bias which is assumed to be a constant value. The compass bias which is mostly dependent on vehicle heading relative to magnetic north (An, 1997) is identified in the EKF using DGPS positions when surfaced. When submerged, the position error covariance grows, but is corrected on surfacing. A relatively short surface time, (for example, 10 seconds) allows the filter to re-estimate biases, correct position estimates and continue with improved accuracy. As a demonstration, the ARIES vehicle was operated in Monterey Bay, June 2000, in a series of runs including a dive-surface-dive-surface sequence. Figure 5, below shows a plot of vehicle position where the solid line to the left indicates the dead reckoning solution and the line to the right represents the EKF solution. In this plot, the vehicle trajectory starts at (0,0) turns counterclockwise underwater and proceeds in a northerly direction for 100 meters, surfaces and takes DGPS measurements, submerges and travels an additional 100 meters and finally surfaces.

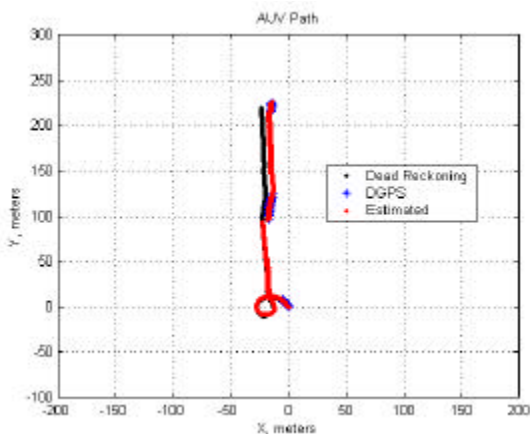


Figure 5. Circular Dive - Underwater Segment - Surface - Dive - Surface Mission. Red Segments are the EKF Solution, Black - The Dead Reckoning Solution, and Blue * are DGPS Values. (Steinspring, 2000)

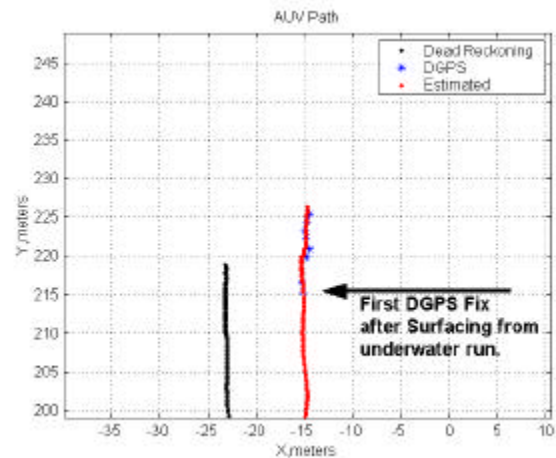


Figure 6. Close up of the Final Surface Showing the Filter Solution together with the DGPS Measurement At the Surface.

In Figure 6, a close up of the final surfacing maneuver shows that there is only a sub meter error in attaining the true DGPS data point as seen by the AshTec G-12 unit on board. The solid line to the left again indicates the dead reckoning solution which is several meters off the mark.

Using the EKF in the Navigation Process always provides a current estimate of vehicle position either surfaced or underwater, and Figure 7 gives an overlay of the EKF solution together with DGPS data from the AshTec unit. Again, sub-meter errors can be seen at the final surfacing position.

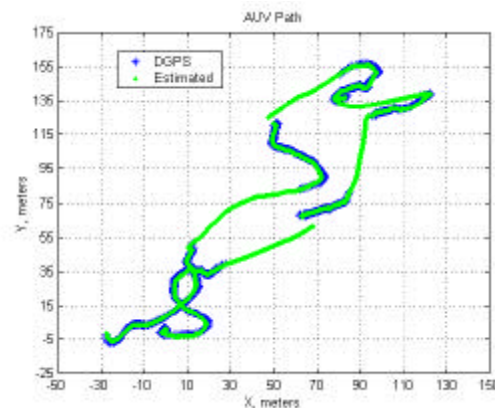


Figure 7. DGPS Data in Blue * with EKF Solution in Green. Segments without the Blue * Correspond to Underwater Segments

VI. SERVER VEHICLE CONCEPT

It is proposed to use the NPS ARIES as a network server vehicle for multi-vehicle cooperative operations. One of

the needs is underwater data transfer between network nodes through noisy communication channels. Use of the server vehicle as a data relay increases the range of communications of the underwater components of the network. Figure 8 describes the concept where in position 1, the ARIES communicates through its acoustic modem with multiple worker vehicles that are engaged in a search pattern. Position 2 shows the ARIES on the surface using a radio modem to report mission status of the worker vehicles (possibly vehicle positions, image snippets of targets, and hydrographic data) to the command ship. While surfaced the server vehicle can receive tactical decision re-tasking commands. Once the new orders are received, the vehicle will submerge and transmit, using its acoustic modem, new tasks to each worker vehicle. Using a server vehicle eliminates the complexity of deploying fixed buoys. Also, a vehicle of this type can achieve close proximity or rendezvous with the worker vehicles allowing for higher acoustic bandwidth data transfer.

Clearly, common communications protocols are needed in order to make this concept viable. Therefore, each vehicle in the network must share a common control language. For instance an agreement could be made to use a set of NEMA type command strings to set waypoints, tracks, behaviors, and status inquiry.

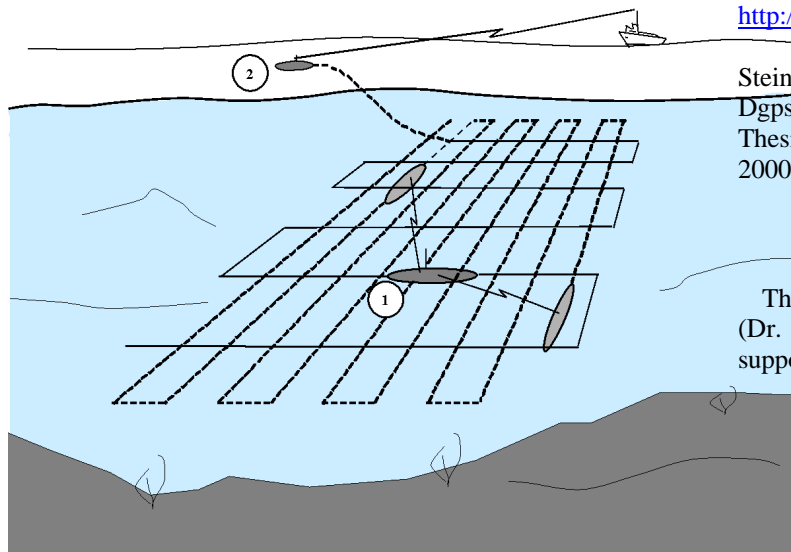


Figure 8. Server vehicle concept. 1. Low bandwidth submerged data transfer between underwater vehicles. 2. High-speed data relay to command ship.

This paper has described a third generation AUV from the NPS Center for AUV Research. A new computer architecture has been described to enable the vehicle to operate as a network server using acoustic and radio communication links. Most importantly, the vehicle has been designed for accurate navigation in shallow water using an extended Kalman filter and DGPS-CP. Research is ongoing in the field of multi-vehicle cooperative behavior and common control languages.

REFERENCES

AUVFest '99, NAVO Report <http://www.cnmoc.navy.mil/auvdemo/index.htm>

Healey, A. J., An, E. A., Marco, D. B., " On Line Compensation of Heading Sensor Bias for Low Cost AUV's", Proceedings of the IEEE Workshop on Autonomous Underwater Vehicles, AUV98, IEEE Catalog. Number 98CH36290, ISBN # 0-7803-5190-8, August 20-21, 1998, Cambridge, Mass. pp 35-42 <http://web.nps.navy.mil/~me/healey/papers/auv98.pdf>

An, P. E., Healey, A. J., Park, J., Smith, S. M., " Asynchronous Data Fusion For AUV Navigation Via Heuristic Fuzzy Filtering Techniques ", Proceedings IEEE, Oceans 97, Halifax, Oct. 1997 IEEE CD-ROM 0-7803-4111-2 http://web.nps.navy.mil/~me/healey/papers/oceans_97.pdf

Steinspring, B. M., " The Experimental Evaluation Of A Dgps Based Navigational Suite In The Aries Auv" MSME Thesis, Naval Postgraduate School, Monterey, CA. June 2000

ACKNOWLEDGEMENTS

The authors wish to thank the Office of Naval Research (Dr. Tom Curtin and Dr. Tom Swean) for the financial support of this project.

VII. CONCLUSIONS